

FIG. 1

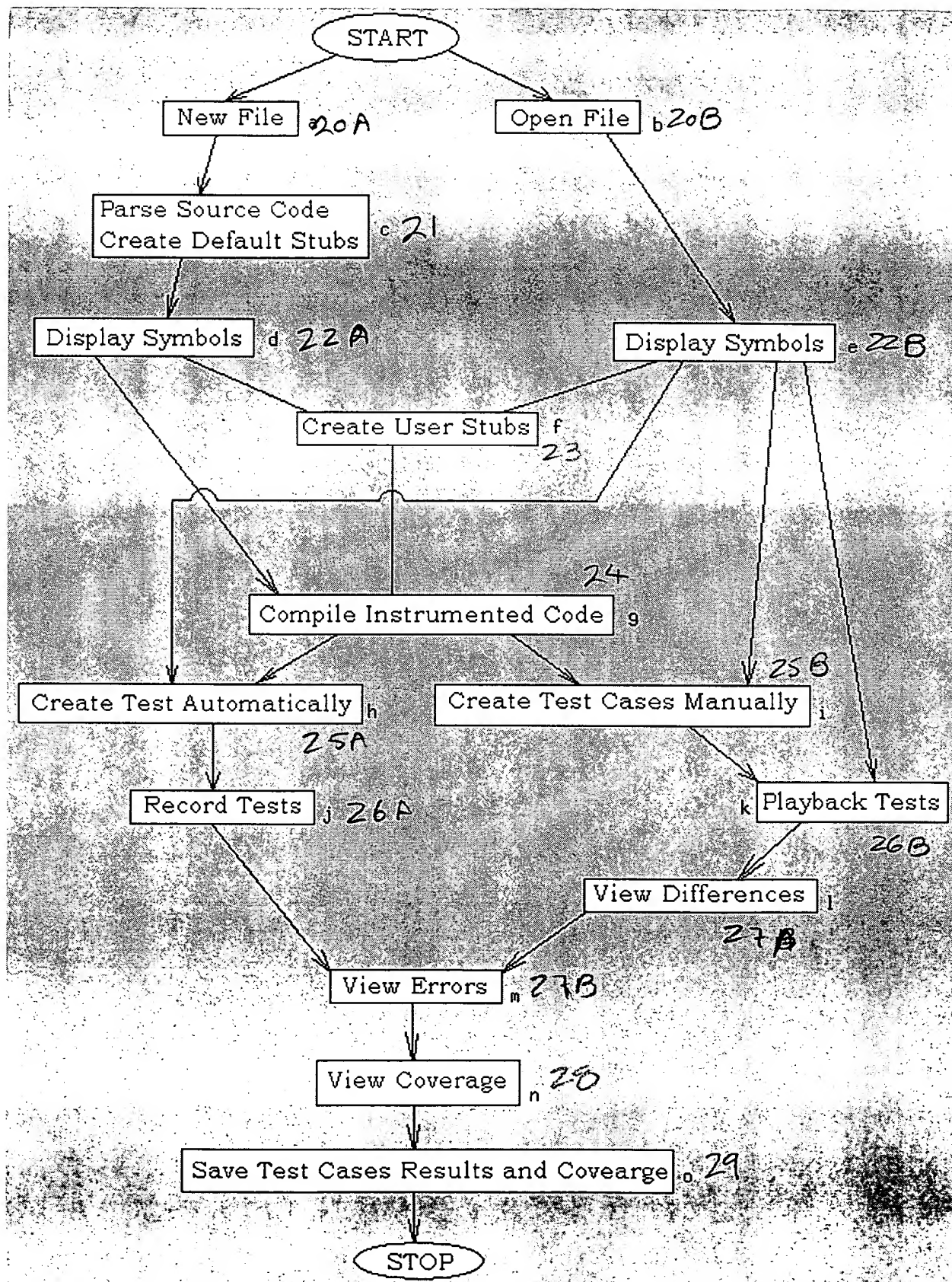


FIG. 2

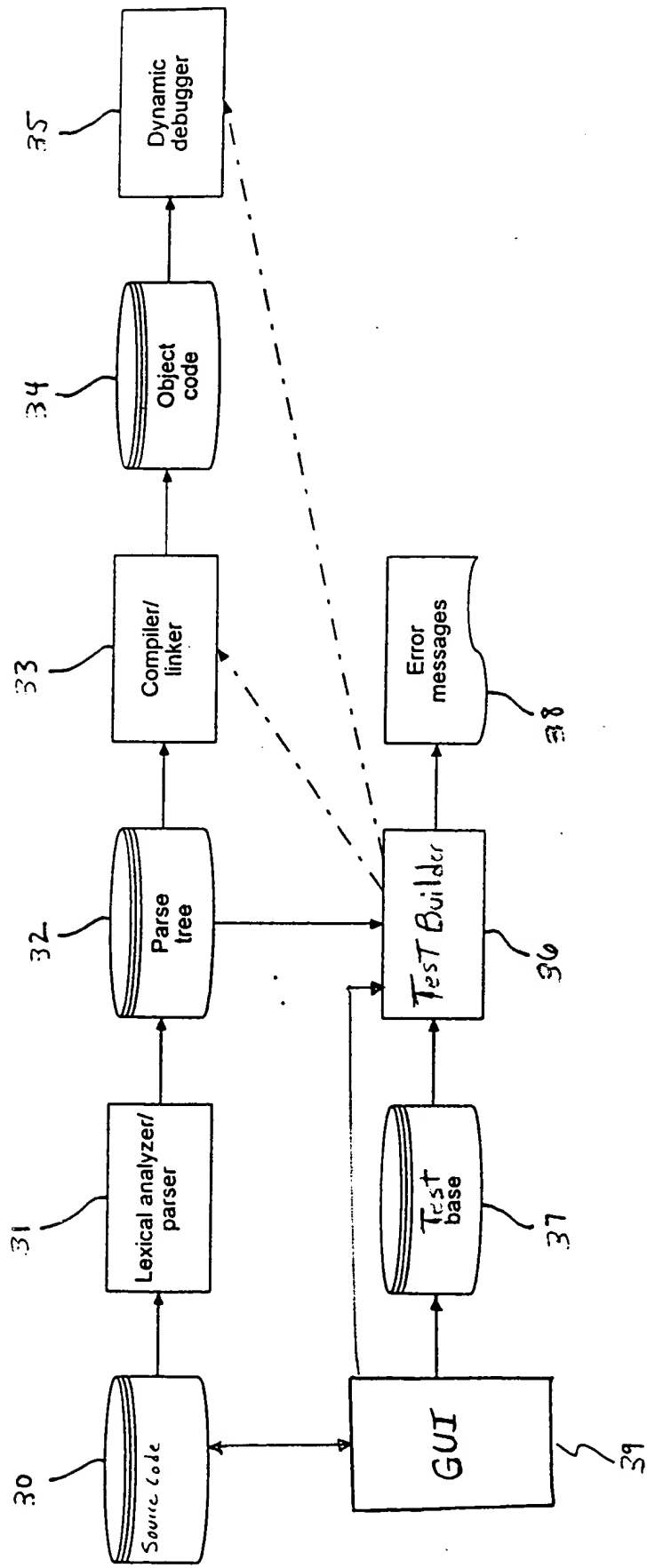


FIG. 3






Button	Name	Action
	Open file(s)	Opens a file chooser from which you can select one or more files. Opened files will be displayed in the File List.
	Read Symbols	Parses the selected file and represents its symbols in the Symbol tree.
	Build Test	Parses the current file, reads its symbols, instruments it if necessary, then compiles and links it.
	Test All	Builds and tests all files currently in the File List.
	Refresh	Recreates your Symbol tree from its current symbol repository, closes all expanded nodes, and deselects the currently selected node. If the file is closed, it will clear the Symbol tree and other tabs.

FIG.4

File List

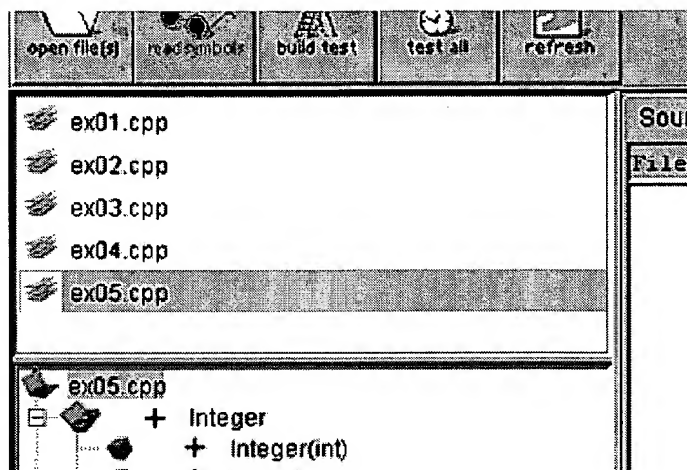


FIG. 5

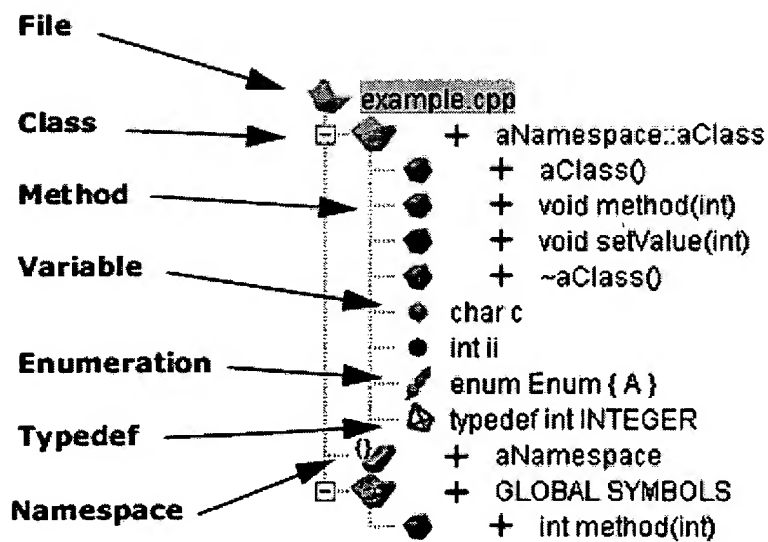


FIG. 6

Source code | Test progress | Results | Test case editor | Symbols | Suppressions

File: C:\cpp\test\examples\ex05.cpp

```
1 // This example illustrates how to test class methods
2 // and functions which have class as a parameter
3
4 class Integer
5 {
6 public:
7     Integer(int i) : _i(i) {}
8     void setInt(int i) { _i = i; }
9     int getInt() { return _i; }
10 private:
11     int _i;
12 }
13
14 int getInt(Integer i)
15 {
16     return i.getInt();
17 }
```

FIG. 7A

Source code | Test progress | Results | Test case editor | Symbols | Suppressions

Tested | Run | Stop | Pause

methods
GLOBAL SYMBOLS
void bubble_sort1(int*, int)
void bubble_sort2(int*, int)
void simple_insertion_sort1(int*, int)
void simple_insertion_sort2(int*, int)
void sort(int*, int)

FIG. 7B

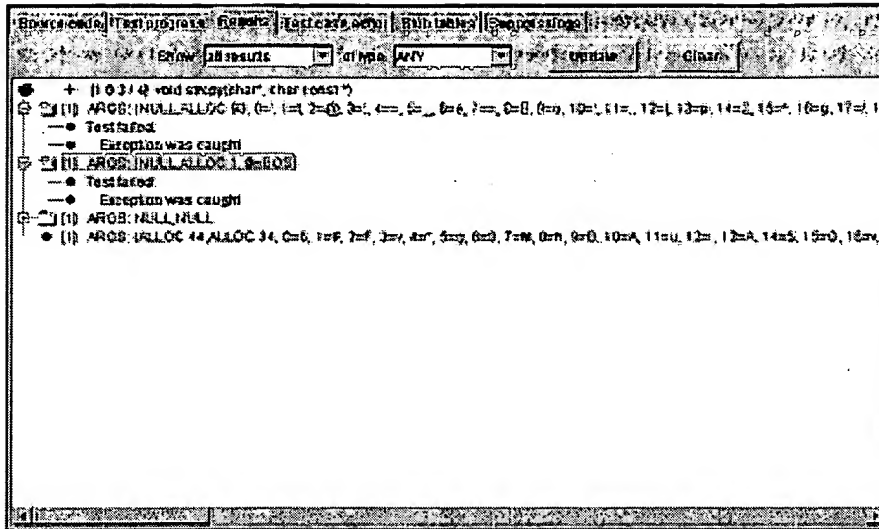


FIG. 7C

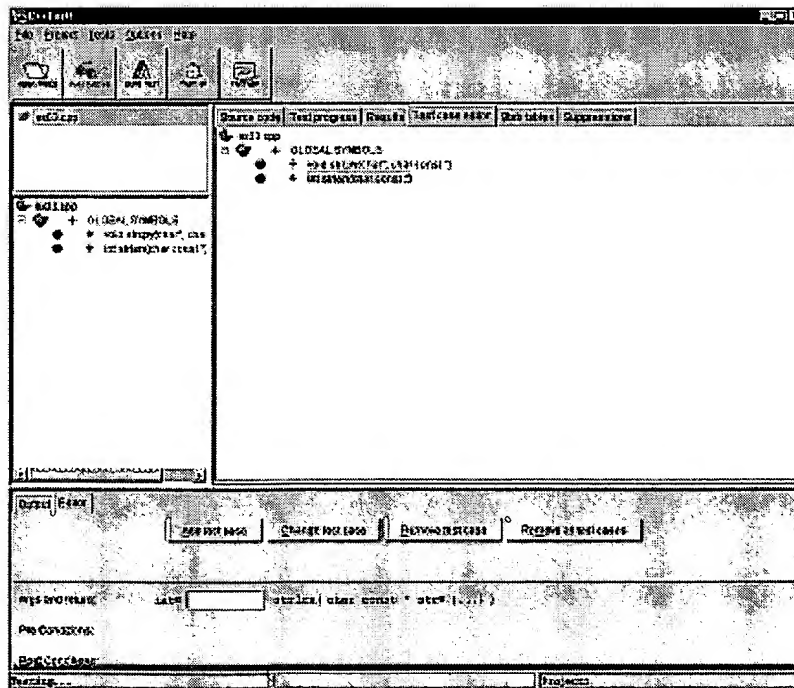


FIG. 7D

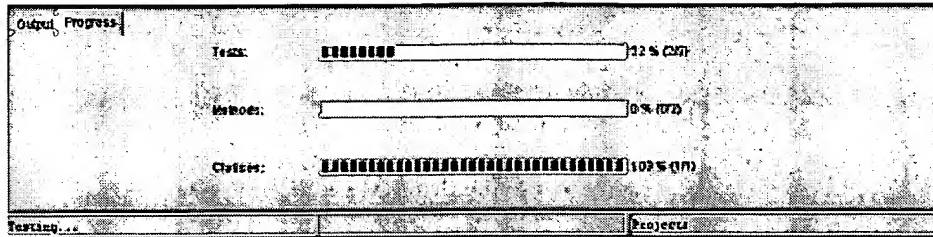


FIG. 7G

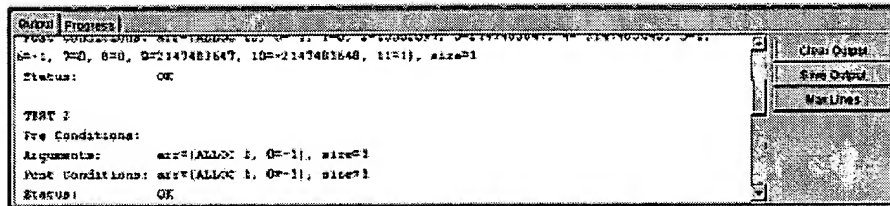


FIG. 7H

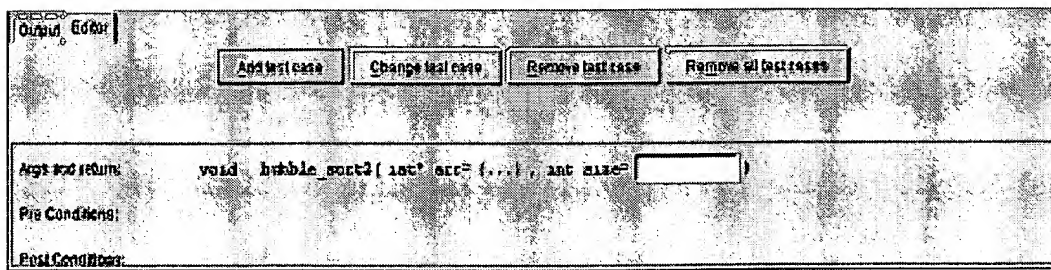


FIG. 7I

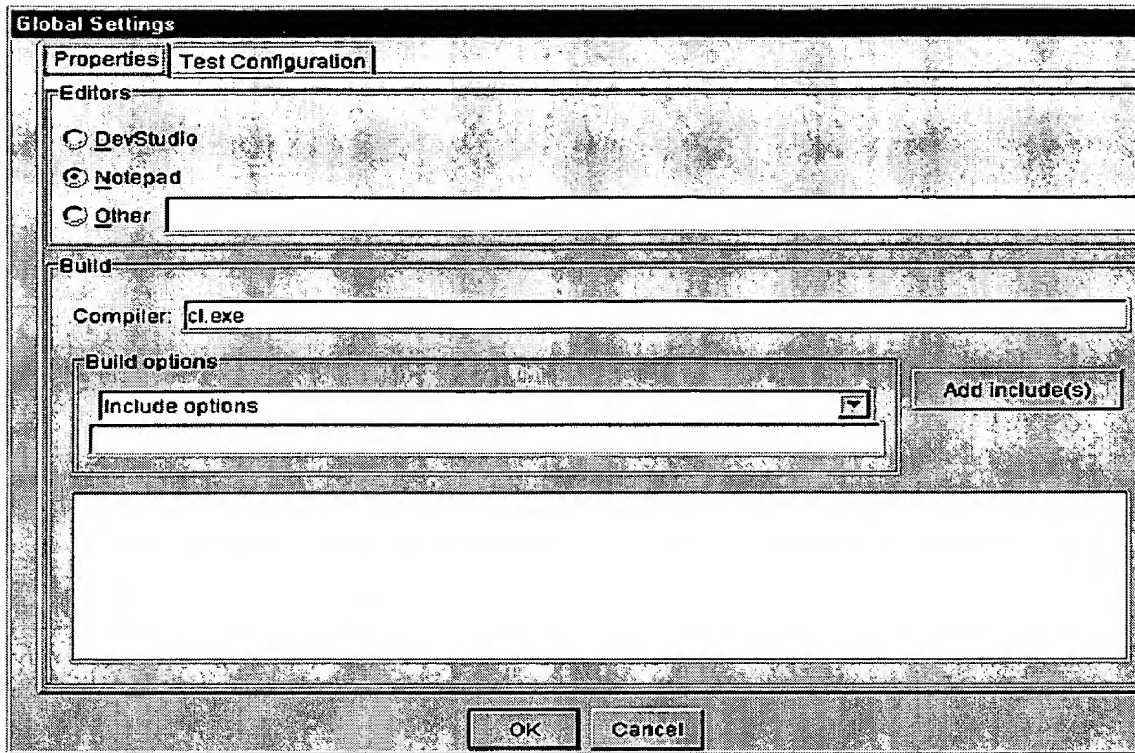


FIG. 7J

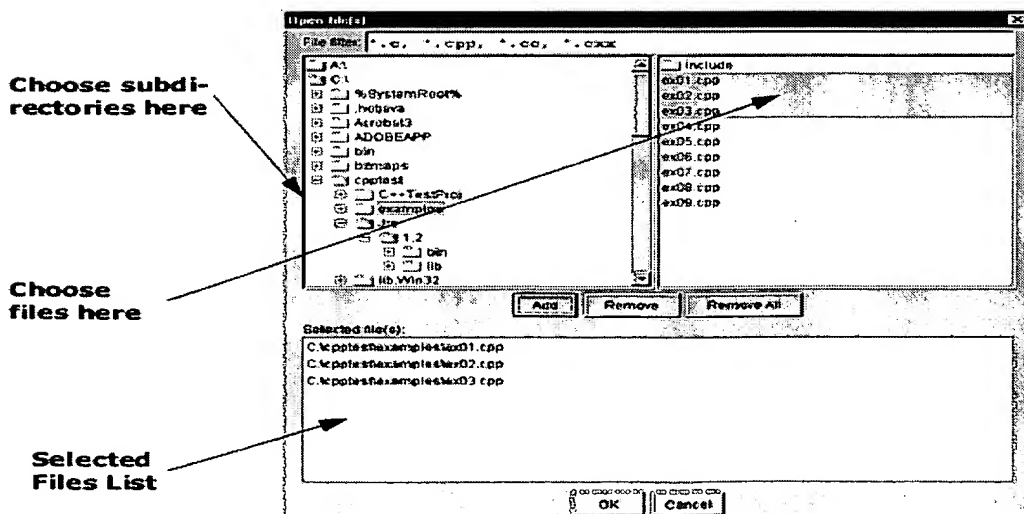


FIG. 8

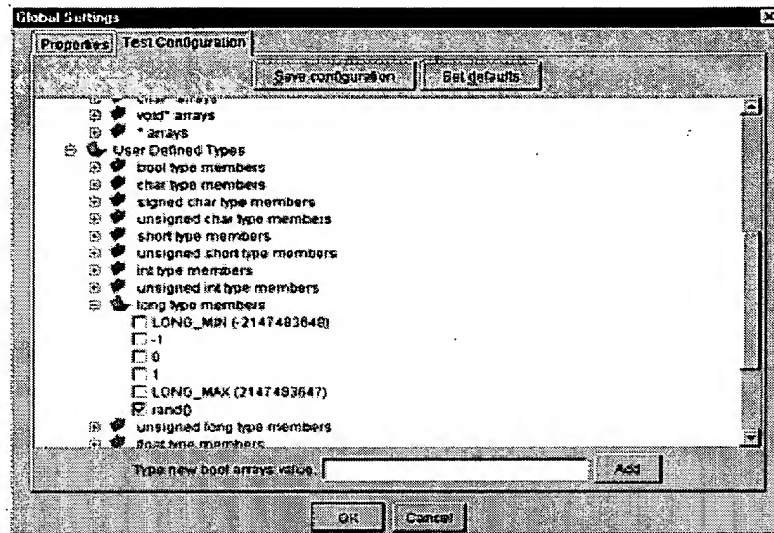


FIG. 9

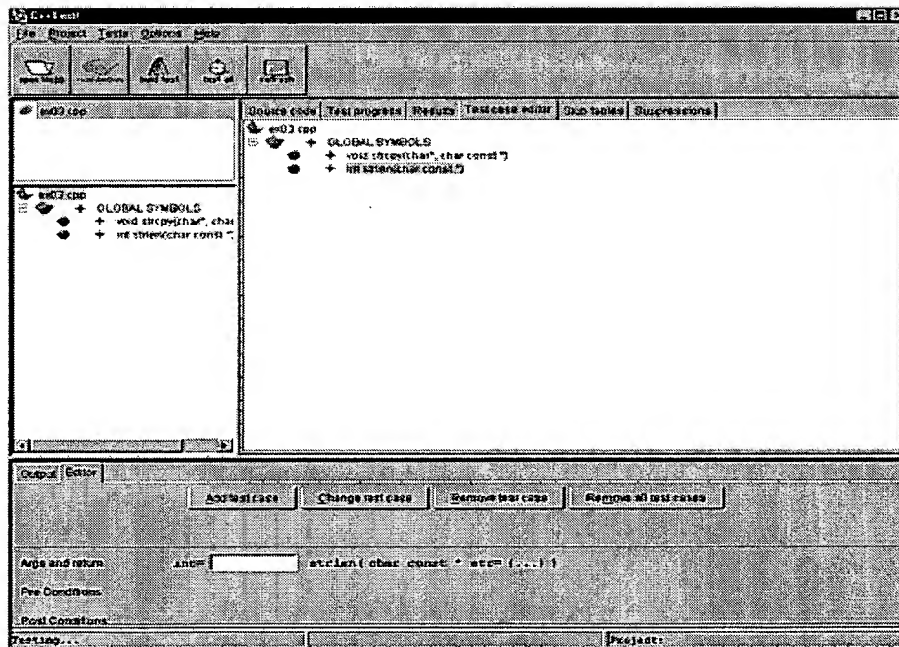


FIG. 10A

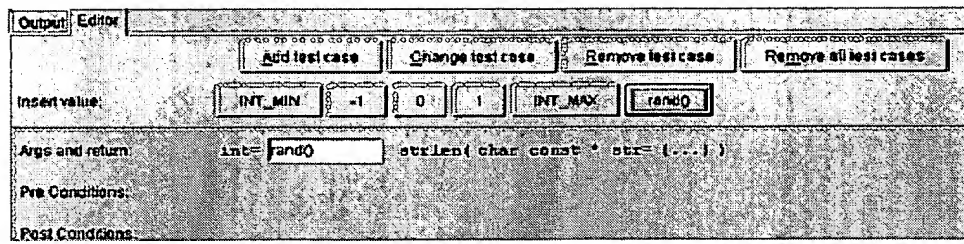


FIG. 10B

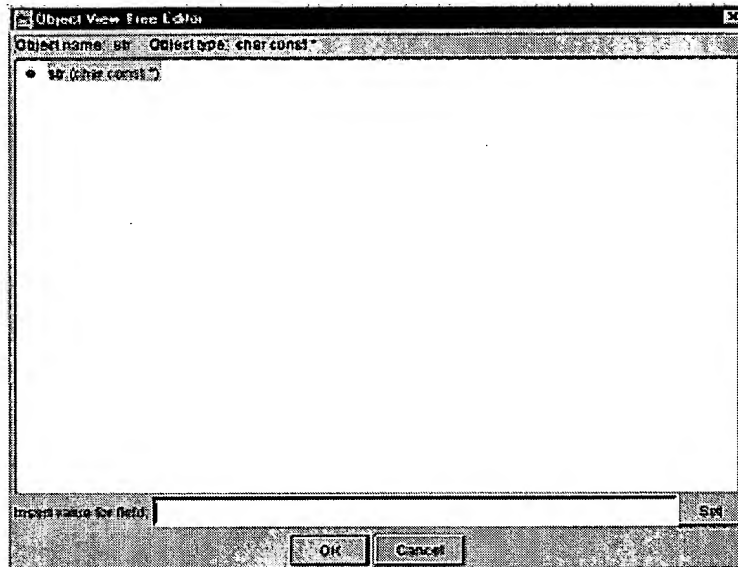


FIG. 10C

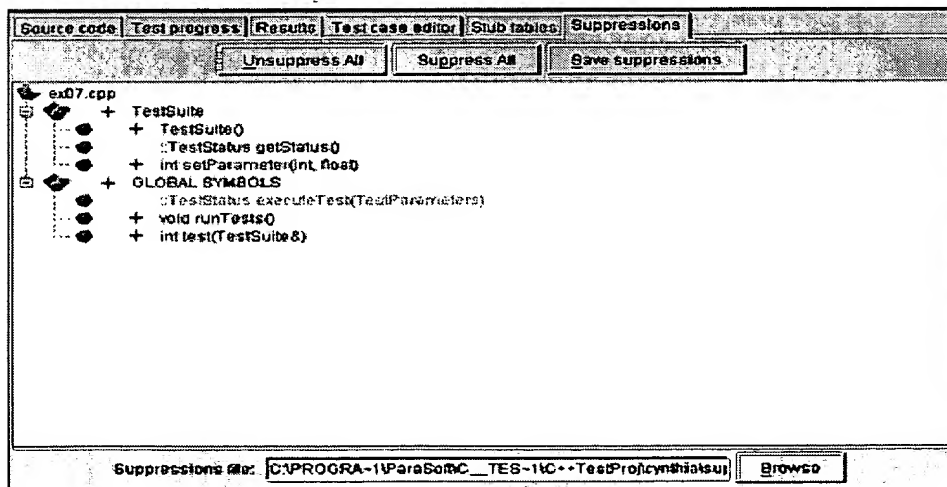


FIG. 11

Line that was already covered

Number of times line was executed

Line currently being executed

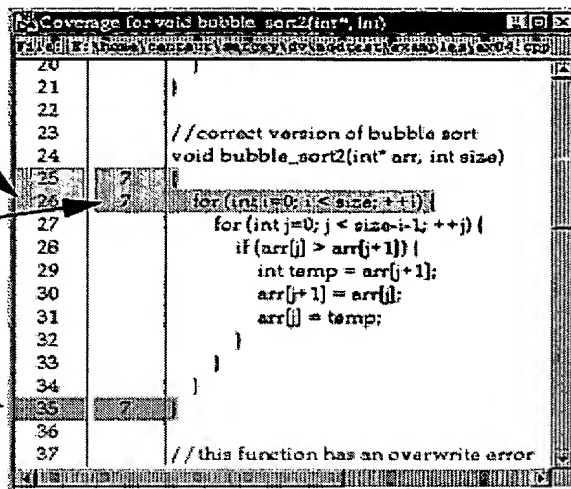


FIG. 12A

Line that was already covered

Number of times line was executed

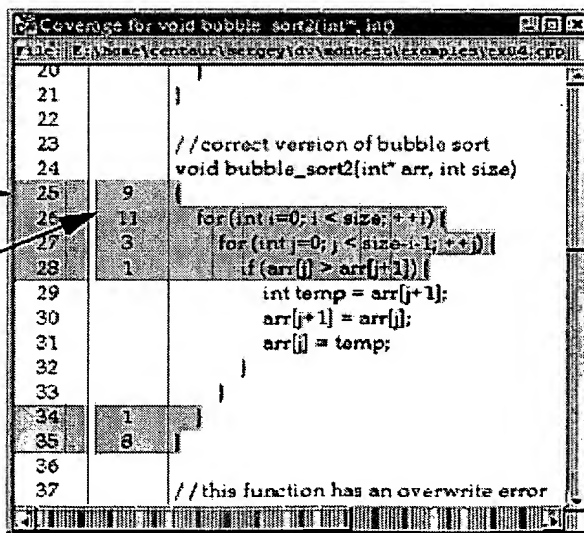


FIG. 12B

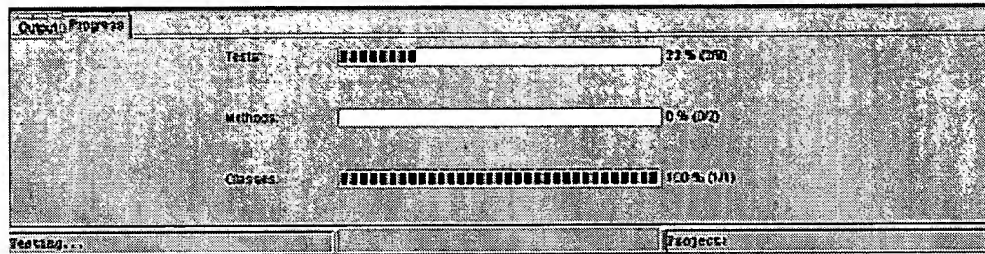


FIG. 13

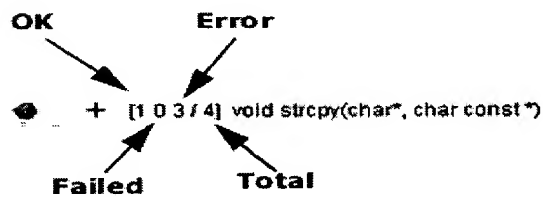


FIG. 14A

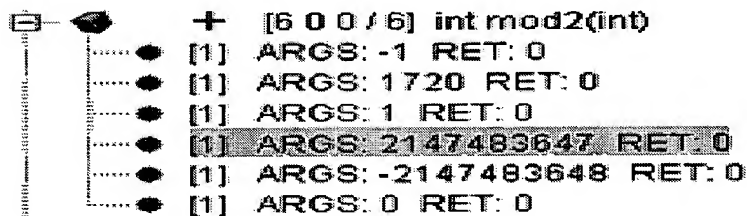


FIG. 14B

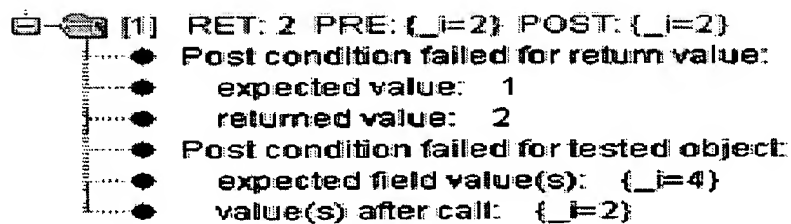


FIG. 14C

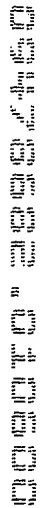


FIG. 14E

